

# 苏州大学课程报告

院、系	计算机学院	年级专业	23 软件工程	姓名	朱金涛	学号	2327406014
课程名称	软件测试与质量保证				成绩		
指导教师	程宝雷	同组实验者	无	实验日期	2025-10-20		

作业名称 作业二 使用 JUnit 进行单元测试

## 一. 目的

- 1、了解测试用例及单元测试的基本概念；
- 2、掌握 Eclipse 中 JUnit 的使用方法；
- 3、掌握 JUnit 中测试用例的编写方法；
- 4、掌握 JUnit 中测试套件的编写方法。

## 二. 内容

参考案例《JUnit 测试》，完成如下内容：

- 1、创建用于测试的程序；
- 2、编写测试用例；
- 3、创建测试套件

## 三. 步骤和结果

### (1) 环境配置

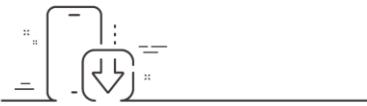
#### ① 安装与准备

- 安装 JDK（版本 21），并配置系统环境变量。

```
C:\Users\zjt20>java --version
java 21.0.7 2025-04-15 LTS
Java(TM) SE Runtime Environment (build 21.0.7+8-LTS-245)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.7+8-LTS-245, mixed mode, sharing)
```

- 安装 Eclipse IDE for Java Developers（2025-09 版）。
- 启动 Eclipse，设置工作空间路径，例如 C:\eclipse-workspace\JUnitDemo。

FOUNDATION



Download Eclipse Technology that is right for you

[Speak at OCX26!](#)  
Share your expertise and shape the future of open source. Submit your talk today!

[Submit your talk!](#)  
[Sponsor OCX](#)



Install your favorite desktop IDE packages

[Learn More](#) [Download](#)

[Download x86\\_64](#) [Download AArch64](#)



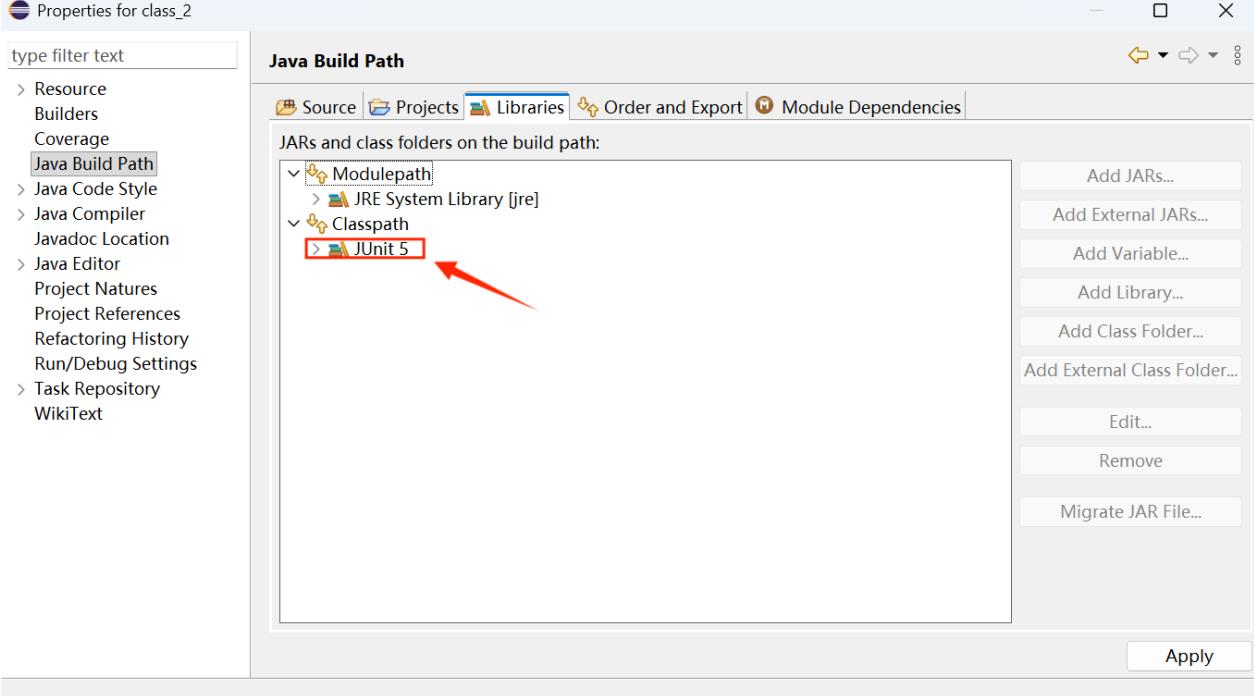
The Eclipse Temurin™ project provides high-quality, TCK certified OpenJDK runtimes and associated technology for use across the Java™ ecosystem

[Learn More](#) [Download](#)

## ② 在 Eclipse 中配置 JUnit 5

- 新建 Java Project → 命名为 JUnitDemo。
- 右键项目 → Build Path → Add Libraries → JUnit → Next。
- 在弹出窗口中选择 JUnit 5 → Finish。
- 若无 JUnit 5 选项，则通过 Help→Install New Software...→输入 <https://download.eclipse.org/releases/latest> 安装“JUnit 5 Support”。
- 验证是否添加成功：展开项目 → 可见“JUnit 5”库。
- 直接删除 module-info.java 以简化配置。



配置结果：  
成功导入 JUnit 5 库并可正常运行单元测试，Eclipse 能识别 JUnit 测试类。

### (2) 创建被测程序

在包 demo 下创建 Caculator.java:

```
package demo;
public class Caculator {
    public int add(int a, int b) {
        return a + b;
    }
}
```

该类提供了一个简单的加法方法，用于测试演示。

结果：程序编译通过。

### (3) 编写测试用例

创建测试类 CaculatorTest.java:

```
package demo;

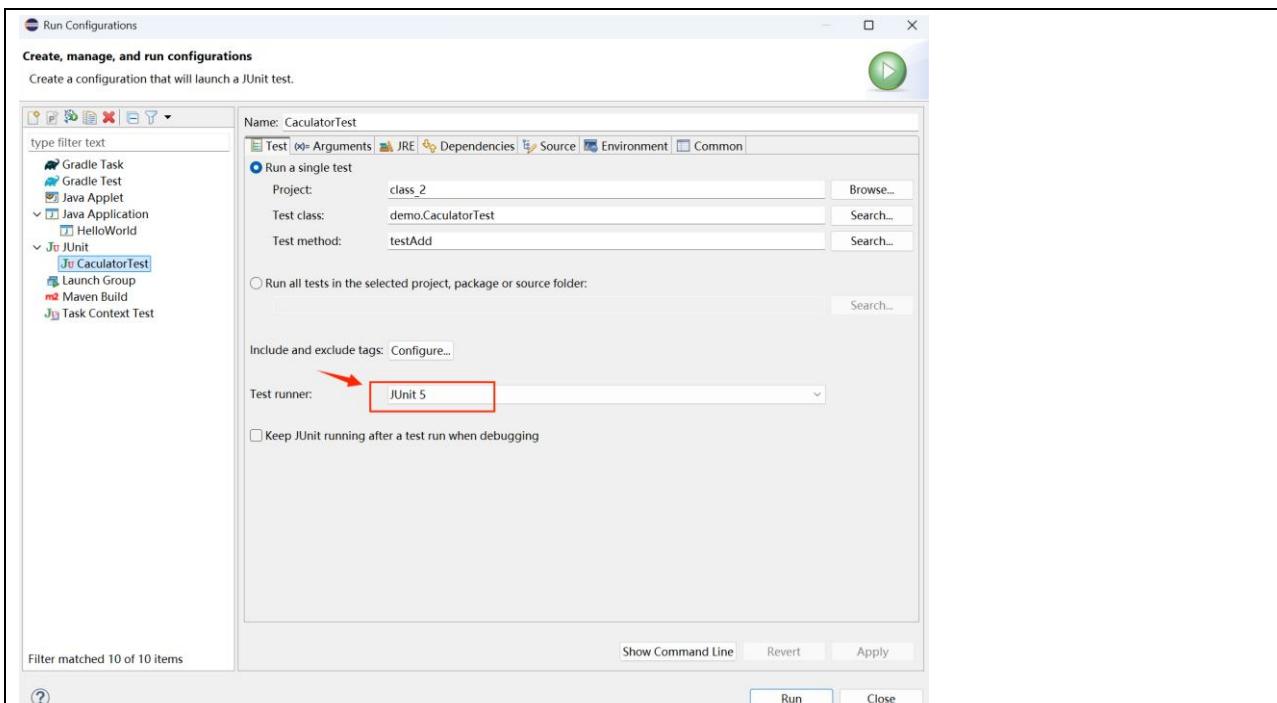
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;

public class CaculatorTest {    // 建议加 public，方便识别

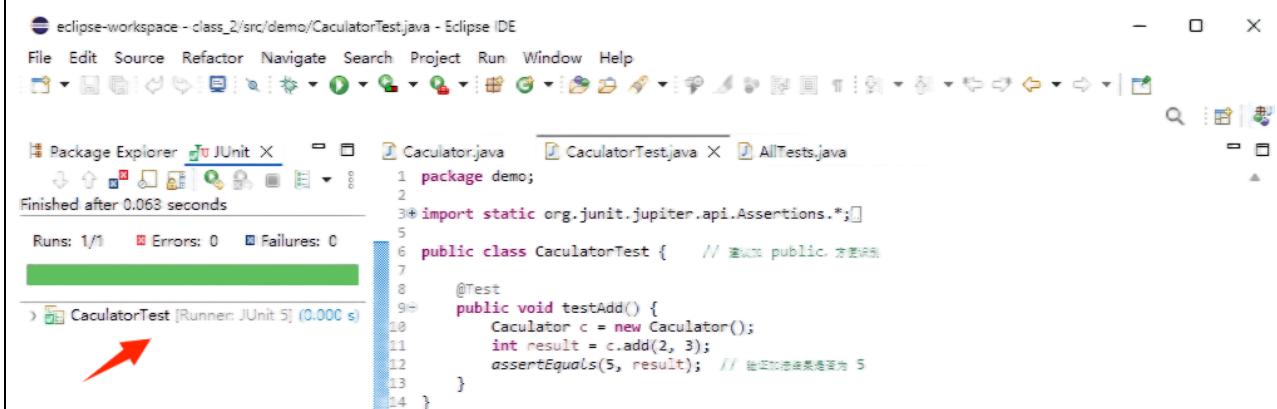
    @Test
    public void testAdd() {
        Caculator c = new Caculator();
        int result = c.add(2, 3);
        assertEquals(5, result); // 验证加法结果是否为 5
    }
}
```

使用@Test 注解定义测试方法，通过断言 assertEquals() 验证输出是否符合预期。

运行方式：右键文件 -> Run As -> Run Configurations



配置之后即可运行，结果如下：



结果：绿色进度条显示测试成功（Runs: 1/1, Errors: 0, Failures: 0）。

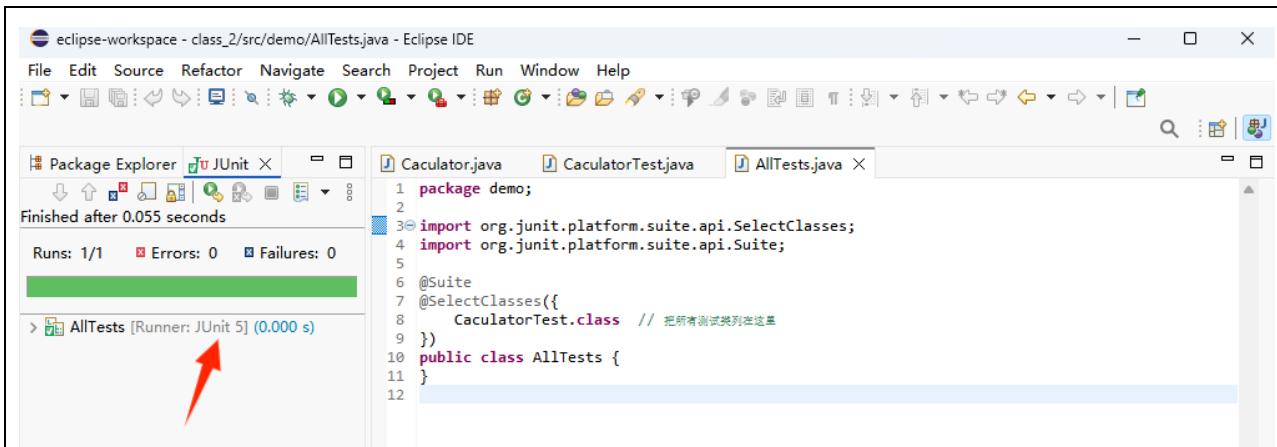
#### (4) 创建测试套件

在同包下创建 AllTests.java:

```
package demo;
import org.junit.platform-suite.api.SelectClasses;
import org.junit.platform-suite.api.Suite;
@Suite
@SelectClasses({
    CaculatorTest.class // 把所有测试类列在这里
})
public class AllTests {
```

右键 → Run As → JUnit Test。

结果： 测试套件成功运行，所有用例通过。



## 四. 总结

本次实验通过在 Eclipse 中使用 JUnit 5 进行单元测试，我深入理解了单元测试在软件开发中的重要作用。实验过程中，从 JDK 与 Eclipse 的安装配置入手，完成了 JUnit 5 的环境搭建和模块依赖问题的解决，使 Eclipse 能够正确识别并运行 JUnit 测试类。随后，创建了 Caculator 被测类，编写了对应的 CaculatorTest 测试用例，利用 @Test 注解和 assertEquals() 断言方法验证加法逻辑的正确性。测试运行结果显示绿色进度条，证明测试通过。最后，通过编写 AllTests 测试套件，实现了对多个测试类的集中管理与一次性运行，进一步掌握了 JUnit 测试组织方式。整个实验帮助我系统地理解了单元测试的基本流程和 JUnit 框架的使用方法，提升了编写高质量、可维护代码的意识与能力。